W522–W525 *Nucleic Acids Research, 2007, Vol. 35, Web Server issue*
doi:10.1093/nar/gkm276

PDB2PQR: expanding and upgrading automated preparation of biomolecular structures for molecular simulations

Todd J. Dolinsky[1], Paul Czodrowski[2], Hui Li[3], Jens E. Nielsen[4], Jan H. Jensen[5], Gerhard Klebe[2] and Nathan A. Baker[1,*]

[1]Department of Biochemistry and Molecular Biophysics, Center for Computational Biology, Washington University in St. Louis, 700 S. Euclid Ave., Campus Box 8036, St. Louis, MO 63110, USA, [2]Department of Pharmaceutical Chemistry, Philipps-University Marburg, Marburg, Germany, [3]Department of Chemistry, University of Nebraska-Lincoln, Lincoln, NE, USA, [4]School of Biomolecular and Biomedical Science, UCD Conway Institute, University College Dublin, Dublin, Ireland and [5]Department of Chemistry, University of Copenhagen, Copenhagen, Denmark

Received January 31, 2007; Revised April 7, 2007; Accepted April 11, 2007

ABSTRACT

Real-world observable physical and chemical characteristics are increasingly being calculated from the 3D structures of biomolecules. Methods for calculating pK_a values, binding constants of ligands, and changes in protein stability are readily available, but often the limiting step in computational biology is the conversion of PDB structures into formats ready for use with biomolecular simulation software. The continued sophistication and integration of biomolecular simulation methods for systems- and genome-wide studies requires a fast, robust, physically realistic and standardized protocol for preparing macromolecular structures for biophysical algorithms. As described previously, the PDB2PQR web server addresses this need for electrostatic field calculations (Dolinsky *et al.*, Nucleic Acids Research, 32, W665–W667, 2004). Here we report the significantly expanded PDB2PQR that includes the following features: robust standalone command line support, improved pK_a estimation via the PROPKA framework, ligand parameterization via PEOE_PB charge methodology, expanded set of force fields and easily incorporated user-defined parameters via XML input files, and improvement of atom addition and optimization code. These features are available through a new web interface (http://pdb2pqr.sourceforge.net/), which offers users a wide range of options for PDB file conversion, modification and parameterization.

INTRODUCTION

Due to the importance of electrostatic interactions in biomolecular systems, a variety of computational methods have been developed for evaluating electrostatic forces and energies [see (1–6) and references therein]. Typical computational electrostatics methods for biomolecular systems can be loosely grouped into two categories: 'explicit solvent' methods, which treat solvent molecules in full molecular detail, and 'implicit solvent' methods, which include solvent–solute interactions in averaged or continuum fashion. Implicit solvent methods are, by definition, limited in detail and therefore lack the atomic-scale accuracy of their explicit solvent counterparts. However, implicit solvent methods have gained increasing popularity, in part due to their elimination of the extensive sampling of solvent configurations required with explicit models (1,3–7).

The basic ingredients of an implicit solvent electrostatics calculation are environmental parameters such as temperature, solvent dielectric and ionic strength; biomolecular atomic coordinates; and parameters for atomic charges and radii. While the environmental parameters are relatively straightforward to specify, the remaining two ingredients can often be difficult to supply. In particular, most biomolecular structures in the Protein Data Bank (PDB) (8) do not contain hydrogen atoms, and many are also missing a fraction of the heavy atom coordinates. The addition of hydrogens and the reconstruction of these missing coordinates is not a trivial process; electrostatic properties obtained from the 'repaired' structures can often be very sensitive to the manner in which missing atoms are added and protonation states are assigned (9,10). Furthermore, inconsistent atomic nomenclature and other

*To whom correspondence should be addressed. Tel: +1-314-362-2040; Fax: +1-314-362-0234; Email: baker@ccb.wustl.edu

force field idiosyncrasies can often make the assignment of atomic charges and radii a cumbersome task. An additional obstacle to the use of PDB structures in electrostatics calculations and other biomolecular computational tasks is the accurate assignment of parameters to 'non-standard' residues and ligands.

Previously (9), we introduced the freely available PDB2PQR service (http://pdb2pqr.sf.net/), which was designed to facilitate the setup and execution of continuum electrostatics calculations from PDB data, particularly by non-experts. The original PDB2PQR server automated many of the common tasks of preparing structures for continuum electrostatics calculations, including adding a limited number of missing heavy atoms to biomolecular structures, estimating titration states and protonating biomolecules in a manner consistent with favorable hydrogen bonding, assigning charge and radius parameters from a variety of force fields, and finally generating 'PQR' output (a PDB-like format with the occupancy and temperature factor columns replaced with charge 'Q' and radius 'R', respectively) compatible with several popular computational biology electrostatics [APBS (10) and MEAD (11)], docking [AutoDock (12)], simulation [AMBER (13)] and visualization [VMD (14), PyMOL (15) and PMV (16)] packages. Since its inception, we have continued to expand the capabilities of the PDB2PQR server to address the challenges associated with ligand parameterization in PDB files and to include several new features.

METHODS

The PDB2PQR web service is driven by a modular, Python-based collection of routines, which provides considerable flexibility to the software and permits non-interactive, high-throughput usage. The service is available via a number of web mirrors listed at http://pdb2pqr.sf.net/. The source code is also available for download from this link, and due to the portability of Python, PDB2PQR can be executed on a wide range of platforms.

Figure 1 outlines the typical workflow of a PDB2PQR job and summarizes the features described in more detail below. The procedures for reconstruction of missing atoms, hydrogen optimization and APBS input generation were described previously (9) and are essentially unchanged in the current version of the software. Since their initial development, these atom reconstruction options have been greatly improved through a number of bug fixes and code optimization, robust support for separate biomolecular chains, and improved chain termini optimization. The following sections describe modified and new elements of the PDB2PQR pipeline.

Titration state assignment by PROPKA

Protonation states for titratable protein groups are assigned by PROPKA 1.0 (http://propka.ki.ku.dk) (17). PROPKA utilizes a very fast empirical method to predict pK_a values and is successful at predicting unusual

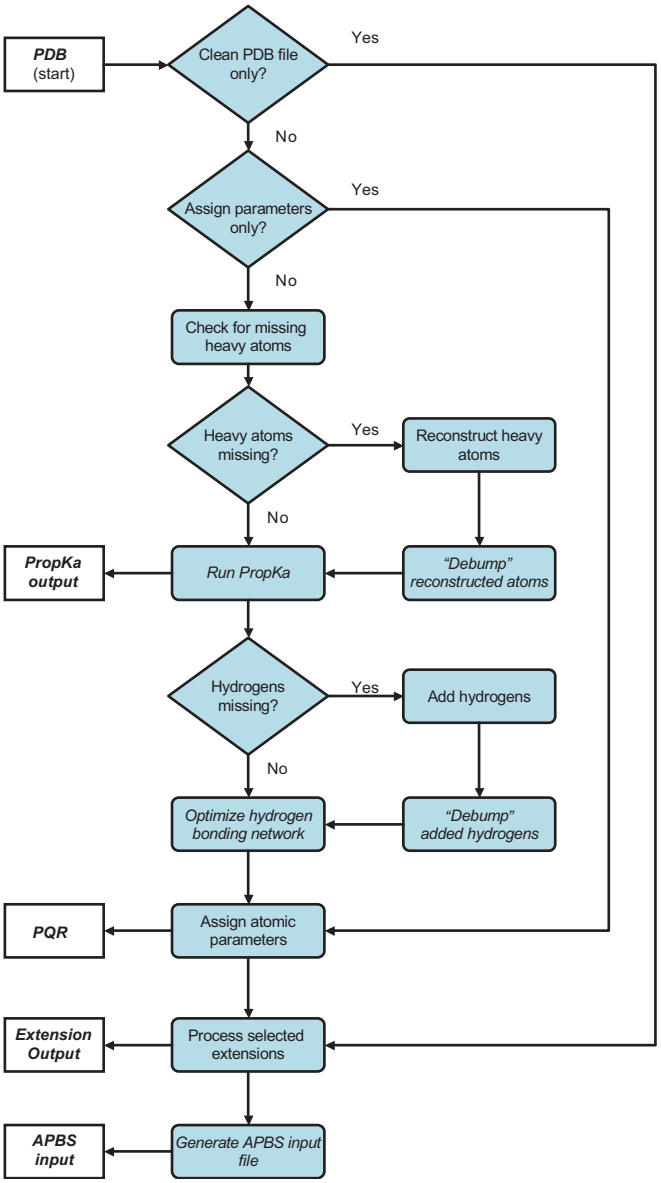

Figure 1. Flowchart demonstrating the sequence of operations performed by the pipeline. The process begins with an input PDB file and ends with a parameterized PQR file and, optionally, an APBS input file.

pK_a values. Recently, a comparative study of several protein pK_a prediction methods showed that PROPKA was the most accurate method overall (18). PROPKA uses a heuristic method to compute the pK_a perturbations due to desolvation, hydrogen bonding and charge–charge interactions. In the current version of PROPKA, contributions from nucleic acids as well as heteroatoms such as bound ions or ligands to the pK_a values are not included. Note that, during the course of titration state assignment, PROPKA generates statistics on residue hydrogen bonding, location and solvent accessibility and Coulombic interactions. This information is available to users as a downloadable text file provided at the end of the PDB2PQR/PROPKA calculation.

Standard residue parameter assignment

PDB2PQR currently allows users to assign protein and (where available) nucleic acid parameters based on explicit solvent AMBER99 (19) and CHARMM27 (20) force fields, the PARSE continuum electrostatics force field (21), a Poisson–Boltzmann-optimized force field by Tan *et al.* (22), or user-defined force fields. User-defined parameters can be uploaded to the PDB2PQR server in a simple flat-file format described in the PDB2PQR user guide. Additionally, PDB2PQR output can be customized to include a variety of atom naming schemes, including AMBER99 (19), CHARMM22 (20), PARSE (21) and an internal naming scheme based on the IUPAC naming recommendations (23). This flexibility in nomenclature was included to facilitate import of PDB2PQR output into other modeling packages. Additionally, the web server provides a 'map' which is output at the end of every PDB2PQR calculation and presents a table of atoms' name/number, residue name, chain name, AMBER atom type and CHARMM atom type to aid in the interpretation of parameter assignment and the development of user-defined charges and radii.

Ligand parameter assignment

The calculation of ligand charges necessitates detailed information on molecular structure and protonation states due to the large variation in the covalent structures of small-molecule protein ligands. The current version of PDB2PQR therefore requires the ligand structure, protonation state and formal charge to be specified by the user in the popular MOL2 (24) format. Ligand structures in MOL2 format are readily available from popular molecular modeling software and free web services such as PRODRG (25). Future versions of PDB2PQR will include a pdb2mol2 parser and automatic assignment of default ligand protonation states from a small-molecule pK_a database.

The calculation of ligand charges in PDB2PQR is based on the partial equalization of orbital electronegativities (PEOE) procedure developed by Gasteiger and Marsili (26). In the PEOE procedure, orbital electronegativities χ are linked to partial atomic charges q by a polynomial expansion ($\chi = a + b \cdot q + c \cdot q^2 + d \cdot q^3$). The coefficients a, b, c and d were optimized by Gasteiger and Marsili using gas phase data on ionization potentials and electron affinities. We utilize a PEOE algorithm, which has been optimized by Czodrowski *et al.* to obtain better agreement between theoretical and experimental solvation energies for a set of small molecules including the polar amino acids (27). The resulting PEOE_PB charges have been tested for small-molecule complexes with trypsin, thrombin (28) and HIV protease (29), and have been found to give results that are in agreement with experimental values.

Post-processing

The current version of PDB2PQR supports an 'extension' directory for user-defined processing of PDB2PQR output. Such extensions might include alternative naming schemes, identification and parameterization of other molecule types, additional hydrogen bond processing, etc. The web servers listed at (http://pdb2pqr.sf.net/) provide only the default PDB2PQR functionality. However, it is straightforward for users to download the PDB2PQR software and setup their own web servers with additional functionality based on custom extensions.

CONCLUSIONS

We have described a number of new features for the free PDB2PQR web server, a service which helps users prepare molecular structures for further computational work by modeling missing atoms, assigning charges and titration states, and providing a mechanism for assignment of ligand parameters. Readers interested in these tasks might also be interested in other servers, which provide complementary services for biomolecular structure processing (30–32). Planned future developments for PDB2PQR include the construction of a pdb2mol2 parser to allow for the automatic parameterization of non-protein atoms, the correct treatment of protein post-translational modification, and the integration of a Poisson–Boltzmann continuum electrostatics-based pK_a calculation algorithm into PDB2PQR. We anticipate that the PDB2PQR service will continue to be a helpful addition to the portfolio of tools available to the structural and computational biology communities.

ACKNOWLEDGEMENTS

N.A.B. and T.J.D. were supported by NIH grant GM069702 and the National Biomedical Computation resource (NIH P41 RR08605); J.H.J. and H.L. were supported by NSF grant MCB 0209941; J.H.J. gratefully acknowledges a Skou Fellowship from the Danish Natural Science Research Council; J.E.N. was supported by a Science Foundation Ireland PIYRA grant (04/YI1/M537); G.K. and P.C. were financially supported by the bilateral CERC3 program of CNRS and DFG (KL 1204/3). The authors would like to thank Andy McCammon for contributions to and support of early versions of the PDB2PQR effort. Funding to pay the Open Access publication charges for this article was provided by NIH grant GM069702.

Conflict of interest statement. None declared.

REFERENCES

1. Baker,N.A. (2005) Improving implicit solvent simulations: a Poisson-centric view. *Curr. Opin. Struct. Biol.*, **15**, 137–143.
2. Darden,T.A. (2001) In Becker,O.M., MacKerell,A.D.Jr, Roux,B. and Watanabe,M. (eds), *Computational Biochemistry and Biophysics*, Marcel Dekker Inc., New York, pp. 91–114.
3. Roux,B. (2001) In Becker,O.M., MacKerell,A.D.Jr, Roux,B. and Watanabe,M. (eds), *Computational Biochemistry and Biophysics.*, Marcel Dekker, New York, pp. 133–152.
4. Davis,M.E. and McCammon,J.A. (1990) Electrostatics in biomolecular structure and dynamics. *Chem. Rev.*, **94**, 7684–7692.
5. Honig,B. and Nicholls,A. (1995) Classical electrostatics in biology and chemistry. *Science*, **268**, 1144–1149.

6. Warshel,A., Sharma,P.K., Kato,M. and Parson,W.W. (2006) Modeling electrostatic effects in proteins. *Biochim. Biophys. Acta Proteins Proteomics*, **1764**, 1647–1676.
7. Roux,B. and Simonson,T. (1999) Implicit solvent models. *Biophys. Chem.*, **78**, 1–20.
8. Berman,H.M., Westbrook,J., Feng,Z., Gilliland,G., Bhat,T.N., Weissig,H., Shindyalov,I.N. and Bourne,P.E. (2000) The protein data bank. *Nucleic Acids Res.*, **28**, 235–242.
9. Dolinsky,T.J., Nielsen,J.E., McCammon,J.A. and Baker,N.A. (2004) PDB2PQR: an automated pipeline for the setup, execution, and analysis of Poisson–Boltzmann electrostatics calculations. *Nucleic Acids Res.*, **32**, W665–W667.
10. Baker,N.A., Sept,D., Joseph,S., Holst,M.J. and McCammon,J.A. (2001) Electrostatics of nanosystems: Application to microtubules and the ribosome. *Proc. Natl. Acad. Sci. USA*, **98**, 10037–10041.
11. Bashford,D. (1997) In Ishikawa,Y., Oldehoeft,R.R., Reynders,J.V.W. and Tholburn,M. (eds), *Scientific Computing in Object-Oriented Parallel Environments*, Springer, Berlin. Vol. 1343, pp. 233–240.
12. Morris,G.M., Goodsell,D.S., Halliday,R.S., Huey,R., Hart,W.E., Belew,R.K. and Olson,A.J. (1998) Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function. *J. Comput. Chem.*, **19**, 1639–1662.
13. Case,D.A., Cheatham,T.E.III, Darden,T., Gohlke,H., Luo,R., Merz,K.M.Jr., Onufriev,A., Simmerling,C., Wang,B. *et al.* (2005) The Amber biomolecular simulation programs. *J. Comput. Chem.*, **26**, 1668–1688.
14. Humphrey,W., Dalke,A. and Schulten,K. (1996) VMD—visual molecular dynamics. *J. Mol. Graph.*, **14**, 33–38.
15. DeLano,W.L. (2002) Palo Alto, CA, The PyMOL Molecular Graphics System.
16. Sanner,M.F. (1999) Python: a programming language for software integration and development. *J. Mol. Graph. Mod.*, **17**, 57–61.
17. Li,H., Robertson,A.D. and Jensen,J.H. (2005) Very fast empirical prediction and rationalization of protein pKa values. *Proteins*, **61**, 704–721.
18. Davies,M.N., Toseland,C.P., Moss,D.S. and Flower,D.R. (2006) Benchmarking pKa prediction. *BMC Biochemistry*, **7**, 18.
19. Wang,J.M., Cieplak,P. and Kollman,P.A. (2000) How well does a restrained electrostatic potential (RESP) model perform in calculating conformational energies of organic and biological molecules? *J. Comput. Chem.*, **21**, 1049–1074.
20. MacKerell,A.D.Jr, Bashford,D., Bellot,M., Dunbrack,R.L.Jr, Evanseck,J.D., Field,M.J., Fischer,S., Gao,J., Guo,H. *et al.* (1998) All-atom empirical potential for molecular modeling and dynamics studies of proteins. *J. Phys. Chem. B*, **102**, 3586–3616.
21. Sitkoff,D., Sharp,K.A. and Honig,B. (1994) Accurate calculation of hydration free energies using macroscopic solvent models. *J. Phys. Chem.*, **98**, 1978–1988.
22. Tan,C., Yang,L. and Luo,R. (2006) How well does Poisson–Boltzmann implicit solvent agree with explicit solvent? A quantitative analysis. *J. Phys. Chem. B*, **110**, 18680–18687.
23. Markley,J.L., Bax,A., Arata,Y., Hilbers,C.W., Kaptein,R., Sykes,B.D., Wright,P.E. and Wüthrich,K. (1998) Recommendations for the presentation of NMR structures of proteins and nucleic acids. *J. Mol. Biol.*, **280**, 933–952.
24. SYBYL Molecular Modeling Software, 7.2 ed. (http://www.tripos.com/mol2/mol2_format3.html); Tripos Inc.: St. Louis, MO, 2006.
25. van Aalten,D.M.F., Bywater,R., Findlay,J.B.C., Hendlich,M., Hooft,R.W.W. and Vriend,G. (1996) PRODRG, a program for generating molecular topologies and unique molecular descriptors from coordinates of small molecules. *J. Comput. Aided Mol. Des.*, **10**, 255–262.
26. Gasteiger,J. and Marsili,M. (1980) Iterative partial equalization of orbital electronegativity—a rapid access to atomic charges. *Tetrahedron*, **36**, 3219–3228.
27. Czodrowski,P., Dramburg,I., Sotriffer,C.A. and Klebe,G. (2006) Development, validation, and application of adapted PEOE charges to estimate pKa values of functional groups in protein-ligand complexes. *Proteins*, **65**, 424–437.
28. Czodrowski,P., Sotriffer,C.A. and Klebe,G. (2007) Protonation changes upon ligand binding to trypsin and thrombin: structural interpretation based on pKa calculations and ITC experiments. *J. Mol. Biol.*, **367**, 1347–1356.
29. Czodrowski,P., Sotriffer,C.A. and Klebe,G. (in press) Atypical protonation states in the active site of HIV-1 protease: A computational study. *J. Chem. Inform. Model.*
30. Gordon,J.C., Myers,J.B., Folta,T., Shoja,V., Heath,L.S. and Onufriev,A. (2005) H++: a server for estimating pKas and adding missing hydrogens to macromolecules. *Nucleic Acids Res.*, **33**, W368–W371.
31. Li,X., Jacobson,M.P., Zhu,K., Zhao,S. and Friesner,R.A. (2007) Assignment of polar states for protein amino acid residues using a interaction cluster decomposition algorithm and its application to high resolution protein structure modeling. *Proteins*, **66**, 824–837.
32. Vriend,G. (1990) WHAT IF: a molecular modeling and drug design program. *J. Mol. Graph.*, **8**, 52–56, 29.